Nucleic Acids Research, 2007, Vol. 35, Web Server issue ***W75–W80***
doi:10.1093/nar/gkm229

Asterias: integrated analysis of expression and aCGH data using an open-source, web-based, parallelized software suite

Ramón Díaz-Uriarte*, Andreu Alibés, Edward R. Morrissey, Andrés Cañada, Oscar M. Rueda and Mariana L. Neves

Statistical Computing Team, Structural and Computational Biology Programme, Spanish National Cancer Center (CNIO), Melchor Fernández Almagro 3, Madrid, 28029, Spain

Received January 8, 2007; Revised March 13, 2007; Accepted March 29, 2007

ABSTRACT

Asterias (http://www.asterias.info) is an open-source, web-based, suite for the analysis of gene expression and aCGH data. Asterias implements validated statistical methods, and most of the applications use parallel computing, which permits taking advantage of multicore CPUs and computing clusters. Access to, and further analysis of, additional biological information and annotations (PubMed references, Gene Ontology terms, KEGG and Reactome pathways) are available either for individual genes (from clickable links in tables and figures) or sets of genes. These applications cover from array normalization to imputation and preprocessing, differential gene expression analysis, class and survival prediction and aCGH analysis. The source code is available, allowing for extention and reuse of the software. The links and analysis of additional functional information, parallelization of computation and open-source availability of the code make Asterias a unique suite that can exploit features specific to web-based environments.

INTRODUCTION

Web-based applications are well suited for the analysis of microarray and genomic data. They do not require the user to install or upgrade any software, the computational capabilities (a concern with the large data sets common in genomic studies) are not limited by the user's hardware (only by the server) and, with the recent advances in web technologies, can offer a user interface and experience very similar to that of desktop applications. Integrated suites that carry out a complete set of analyses of several different types of data can be very appealing for many users, as the applications within the suite present a similar interface, have homogeneous input requirements and allow the analysis of various types of data that many wet-lab researchers deal with routinely (e.g., from microarray data normalization to aCGH). In addition, web-based tools offer the opportunity to quickly bring new methodological developments to many potential users. Therefore, there is room for additional work in integrated web-based suites to incorporate key statistical and methodological advances.

Web-based tools: requirements and desirable features

Web-based tools do not need to compromise on statistical rigor and can use validated and state-of-the-art methods. When trying to discover differentially expressed genes, multiple testing problems should be taken into account (1,2) and, since many microarray studies are really observational studies with human patients, it is often necessary to include additional clinical covariates to minimize confounding problems (3,4). In addition, we can also borrow information from all genes in the array when carrying out the test for each gene, using moderated statistics and Empirical Bayes approaches (5). When dealing with classification and prediction, it is crucial to avoid biases that lead to overoptimistic estimates of the error rates. These biases include "selection bias" (6,7) and bias caused by selecting and reporting the error rate of the classifier (among a set of classifiers) with the smallest cross-validated error rate (8,9). Additionally, gene selection in the context of classification often yields many solutions with similar prediction errors, but which share few common genes (10–12); being unaware of the possible instability of our results can lead to a false sense of certainty that the given set is special and distinct.

*To whom correspondence should be addressed. Tel: +34 91 224 6900; Fax: +34 91 224 6972; Email: rdiaz02@gmail.com
Present address:
Andreu Alibés, Design of Biological Systems Group, Center for Genomic Regulation, Dr. Aiguader 88, 08003 Barcelona, Spain

In addition to statistical rigor, a modern tool should incorporate the increasing availability of multicore processors and clusters built with off-the-shelf components, which are probably the major opportunities for significant performance gains in the near future (13,14). MPI (15) is one approach to parallelize computations over several CPUs and/or processor cores, thus decreasing execution time. Interestingly, web-based applications are well suited for this task; if deployed in a computing cluster, the parallelization, while transparent for the user, permits harvesting computational resources that are rarely available to individual researchers.

To help in the interpretation of results (16,17), web-based tools are ideally suited to link to additional sources of information, such as PubMed references, gene ontology (GO) terms, and the UCSC and Ensembl databases and KEGG and Reactome pathways. Moreover, it is possible to carry out further analysis with this additional information, such as highlighting features (e.g. pathways, GO terms, etc.) that might be characteristic of a set of selected genes that are, say, very common among the genes that tend to be repeatedly selected as relevant for a classification problem. This usage of additional information can help us understand whether there are biological commonalities behind the possible multiple solutions (see above).

Finally, the availability of source code, under an open-source license, allows other researchers to further improve the method and provide bug fixes, use the code for instruction and teaching, permits to verify claims by method developers, encourages reproducible research, and ensures that the international research community remains the owner of the tools it needs to carry out its work (18). These features facilitate fast methodological development based on previous work, and expedite the transfer of results to applied research. The value of the source code is further enhanced if best practices (19) as well as common open-source practices (including public code repositories and open bug tracking) are followed, ultimately allowing the building of a community of contributors (20).

ASTERIAS: UNIQUE FEATURES

Some of the currently available web-based suites include RACE (26), MIDAW (27), Gepas (28) and CARMAweb (29). All of these, however, fail one or more of the above requirements. We have thus developed Asterias to fulfill those requirements. First, Asterias is the only web-based application which we know of that is designed, from the beginning, to make extensive use of parallelization in its computations. The speed up can be dramatic when run in a computing cluster (in our own installation of 30 dual-processor servers, some applications speed up by factors of 30 × to 50 ×). Second, Asterias, as with some other suites, includes tools that cover the complete range of needs of many researchers (from normalization to aCGH analysis, including imputation, differential expression and class prediction), but Asterias is the only suite that includes tools for searching for large sets of predictive genes (GeneSrF), and gene selection, molecular signatures and prediction with survival data (SignS). Third, we provide statistically rigorous and state-of-the-art methods, from the well-known BioConductor limma package (5), in the study of differential expression, to the best available methods for aCGH analysis, as reported in recent reviews (30,31). Moreover, we facilitate the analysis of multiple solutions in class prediction and gene selection tools (e.g. frequency of genes in bootstrap and cross-validation runs and similarity of solutions with regards to biological role via an analysis of additional information—see below). Fourth, the development of Asterias includes functional and regression testing of our applications, using publicly available and open-source tests; this is also a unique feature of Asterias.

In addition, the newest release of Asterias includes two important additions. We make (virtually) all of our code available under open-source licenses (GNU GPL and Affero GPL) and have an open-source development mode, including open bug tracking and full repository history available. Finally, an important novelty with respect to our latest release, the user can analyze the results (e.g. the genes that have been selected as good prognosis classifiers) and examine PubMed references, GO terms, KEGG pathways or Reactome pathways for those genes using the new PaLS web server. PaLS, coupled with the examination of multiple solutions, can ease the biological interpretation of the results, specially in studies of gene selection and classification.

Asterias shares some common history with the GEPAS suite (28), and one of the authors of Asterias (RD-U) was heavily involved in the development of GEPAS (32–34) and related tools (35–37). Nowadays, Asterias and GEPAS only share the tool DNMAD—although the R code in Asterias' DNMAD has changed to adapt it to the latest BioConductor releases— and a similar approach to web server load-balancing, via Pound or LVS, with everything else being different. A brief history of the split can be found at http://asterias.bioinfo.cnio.es/Asterias.Gepas.html. The main differences between Asterias and GEPAS are our strong commitment to parallel computing, differences in the type of applications being developed (e.g. SignS, ADaCGH, GeneSrF, PomeloII) and software development mode (all of our code is available under open-source licenses, including complete repositories and functional tests).

FUNCTIONALITY, INPUT, OUTPUT

Figures 1 and 2 show the main functionality provided by each of the Asterias applications, the relationships between the tools and the main input and output of each application. All the analysis tools are accessible from preP, but can also be accessed directly, and preP can be accessed either directly or from DNMAD.

Input to all applications are plain text files, with tab-separated columns. Further details are provided in the online help of each application. Output of most applications includes both text-like output, with clickable links to IDClight (38) and PaLS and graphical output. Some applications (e.g. IDconverter) can also

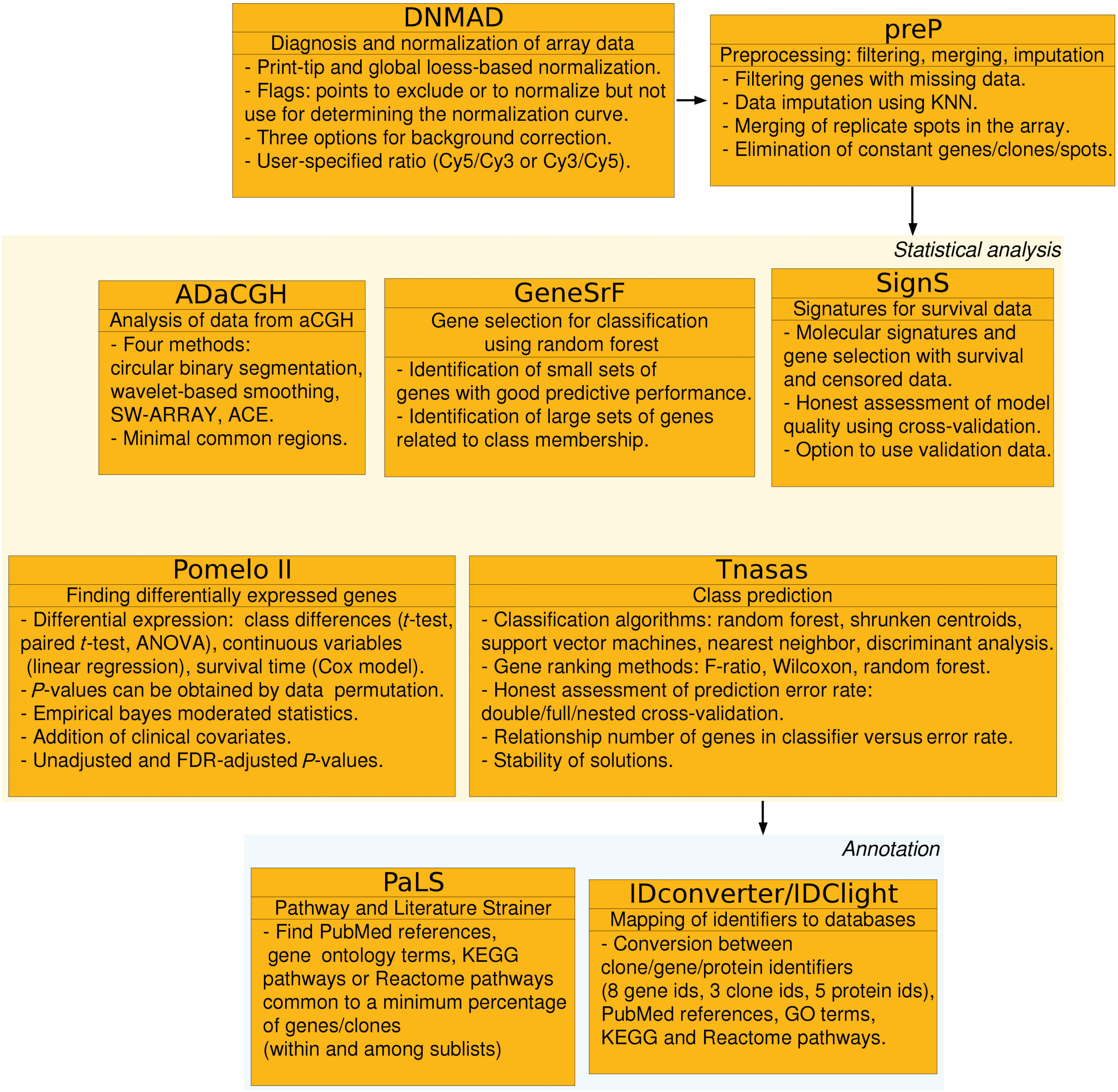
DNMAD
Diagnosis and normalization of array data
- Print-tip and global loess-based normalization.
- Flags: points to exclude or to normalize but not use for determining the normalization curve.
- Three options for background correction.
- User-specified ratio (Cy5/Cy3 or Cy3/Cy5).
preP
Preprocessing: filtering, merging, imputation
- Filtering genes with missing data.
- Data imputation using KNN.
- Merging of replicate spots in the array.
- Elimination of constant genes/clones/spots.
Statistical analysis
ADaCGH
Analysis of data from aCGH
- Four methods: circular binary segmentation, wavelet-based smoothing, SW-ARRAY, ACE.
- Minimal common regions.
GeneSrF
Gene selection for classification using random forest
- Identification of small sets of genes with good predictive performance.
- Identification of large sets of genes related to class membership.
SignS
Signatures for survival data
- Molecular signatures and gene selection with survival and censored data.
- Honest assessment of model quality using cross-validation.
- Option to use validation data.
Pomelo II
Finding differentially expressed genes
- Differential expression: class differences (t-test, paired t-test, ANOVA), continuous variables (linear regression), survival time (Cox model).
- P-values can be obtained by data permutation.
- Empirical bayes moderated statistics.
- Addition of clinical covariates.
- Unadjusted and FDR-adjusted P-values.
Tnasas
Class prediction
- Classification algorithms: random forest, shrunken centroids, support vector machines, nearest neighbor, discriminant analysis.
- Gene ranking methods: F-ratio, Wilcoxon, random forest.
- Honest assessment of prediction error rate: double/full/nested cross-validation.
- Relationship number of genes in classifier versus error rate.
- Stability of solutions.
Annotation
PaLS
Pathway and Literature Strainer
- Find PubMed references, gene ontology terms, KEGG pathways or Reactome pathways common to a minimum percentage of genes/clones (within and among sublists)
IDconverter/IDClight
Mapping of identifiers to databases
- Conversion between clone/gene/protein identifiers (8 gene ids, 3 clone ids, 5 protein ids), PubMed references, GO terms, KEGG and Reactome pathways.

Figure 1. Asterias: functionality and data and information flow between sets of applications (see details in Figure 2). References for ADaCGH methods are: circular binary segmentation (21), wavelet-based smoothing (22), SW-ARRAY (23) and ACE (24). The method implemented in SignS is from (25).

provide tabular output in other formats (e.g. Microsoft Excel). Screenshots of output are provided in the Supplementary Data.

IMPLEMENTATION

Most of the statistical functionality is written in R (39), with some code in C/C++ (Pomelo II and several dynamically loadable code in R packages), and extensive use of parallelization using MPI and R interfaces to MPI. The R code uses standard R or BioConductor packages (some of them modified to allow parallel computation) and our own packages (e.g. varSelRF, ADaCGH). Full details on the R and BioConductor packages used are provided in the help pages of each application. The web interfaces and input data validation are written in Python (with some legacy Perl and PHP in DNMAD and IDconverter). Clickable figures and tables are usually generated using R, with additional post-processing

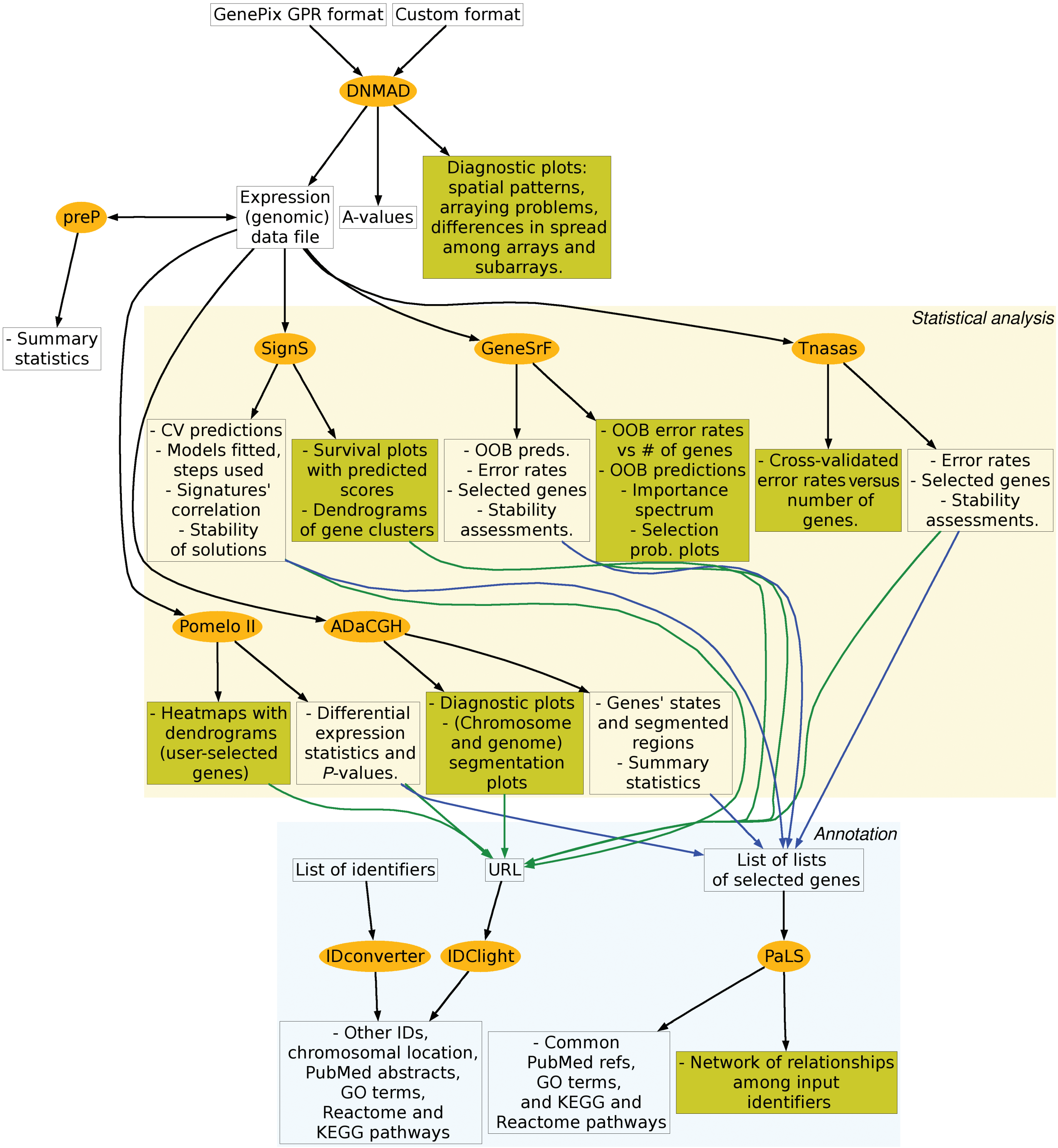
GenePix GPR format
Custom format
DNMAD
Diagnostic plots: spatial patterns, arraying problems, differences in spread among arrays and subarrays.
preP
Expression (genomic) data file
A-values
- Summary statistics
Statistical analysis
SignS
GeneSrF
Tnasas
- CV predictions - Models fitted, steps used - Signatures' correlation - Stability of solutions
- Survival plots with predicted scores - Dendrograms of gene clusters
- OOB preds. - Error rates - Selected genes - Stability assessments.
- OOB error rates vs # of genes - OOB predictions - Importance spectrum - Selection prob. plots
- Cross-validated error rates versus number of genes.
- Error rates - Selected genes - Stability assessments.
Pomelo II
ADaCGH
- Heatmaps with dendrograms (user-selected genes)
- Differential expression statistics and P-values.
- Diagnostic plots - (Chromosome and genome) segmentation plots
- Genes' states and segmented regions - Summary statistics
Annotation
List of identifiers
URL
List of lists of selected genes
IDconverter
IDClight
PaLS
- Other IDs, chromosomal location, PubMed abstracts, GO terms, Reactome and KEGG pathways
- Common PubMed refs, GO terms, and KEGG and Reactome pathways
- Network of relationships among input identifiers

Figure 2. Asterias: input/output and data and information flow between applications. Black and blue arrows involve files, green arrows URLs. Olive boxes denote graphical output.

using Python. The database server for IDconverter, IDClight and PaLS is MySQL. Scripts for database management and generation are also written in Python. JavaScript is used in several applications, most notably in Pomelo II (AJAX), but also on clickable figures and collapsible trees. Booting and halting the LAM/MPI universes is accomplished by a combination of Python and shell scripts. We create a new LAM/MPI universe for each run of each application, and the actual nodes/CPUs that are used in a LAM/MPI universe are determined at run-time (thus excluding nodes that are down).

Documentation, help, bug tracking

Online help, including tutorials, examples and sample files, is available for all applications. Pomelo II includes additional tutorials as flash movies. The online tutorials and examples are licensed under a Creative Commons license (http://www.creativecommons.org), allowing for redistribution and classroom use. The R packages have, additionally, help available in the standard R format. Bug tracking is available from the Bioinformatics.org project page http://bioinformatics.org/bugs/?group_id=630.

Availability

Our publicly accessible installation runs on a cluster with 30 dual-CPU nodes with Debian GNU/Linux. The web service is load-balanced (we are currently using Linux Virtual Server, but have used Pound in the past), which ensures balancing of the master nodes for MPI and of the non-parallelized applications (e.g. preP). All of the code (except, temporarily, for PaLS) is available under open-source licenses (either GNU GPL v.2 or Affero Public License). The complete repositories can be downloaded from Bioinformatics.org (http://bioinformatics.org/asterias) or Launchpad (https://launchpad.net/asterias). The R package varSelRF is also available from the R repositories.

Testing, maturity and number of accesses

Asterias includes a test suite that uses FunkLoad (http://funkload.nuxeo.org). The test suite tests the user interface, handling of error conditions and incorrectly formated files and the numerical output, and can be run on demand, and wherever new changes are introduced in the software, thus ensuring appropriate quality control and regression testing. The complete code is also available (see "Functional testing" in the repositories). For Pomelo II (which makes extensive use of AJAX), additional tests using Selenium (http://www.openqa.org/selenium/) are available (http://pomelo2.bioinfo.cnio.es/tests.html); these tests verify that the application runs correctly under different operating systems and browsers.

Asterias is a mature suite. Its oldest application, DNMAD (40), has been running since October 2003, and the newest one, PaLS, has been running since October 2006. The rest of the applications have been running for at least a year, often considerably longer. The number of data sets analyzed (note that these are counts of actual numbers of successfully uploaded files, not just hits) in the 10-month period February 1, 2006 and November 30, 2006, range from 3700 and 2900 for preP and Pomelo II, respectively, between to 530 and 340 for SignS and GeneSrF, except for IDconverter and IDClight, which have over 70 daily uses.

FUTURE WORK

Our main development effort is focused on making Asterias easy to install and deploy, from laptops to clusters of workstations. We are currently re-implementing all of Asterias using Pylons (http://pylonshq.com), a Python web framework, together with installation scripts that ease the configuration, management and monitoring of the computing nodes and parallel computing layers. We are also exploring other languages and paradigms, such as QHTML (41), built on top of Mozart/Oz, to solve the problem that 'Building web-based applications requires the mastering of a number of languages/technologies (e.g. HTML, CSS, CGI, ASP, PHP, XML, etc.). Such languages and technologies were created to address different aspects on a by-need, evolutionary manner. The result is a plethora of tools that are fitted together in an *ad hoc* fashion'. (41).

In both cases, our ultimate objective is developing a general framework (or at least a large enough set of case examples) that will make it much simpler for any bioinformatician/biostatistician to take new ideas and developments from the primary methodological research and make them quickly available as web-based applications. These web-based applications should be capable of using advances in computing and hardware (multicore CPUs, computing clusters built with off-the-shelf components, parallel computing and concurrency) and web technologies (e.g., AJAX).

SUPPLEMENTARY DATA

Supplementary Data are available at NAR Online.

ACKNOWLEDGEMENTS

We want to thank the many testers at CNIO and elsewhere for feedback on the applications and for bug reports. Bionformatics.org and Launchpad provided hosting for the repositories. Our applications would not have been possible without the excellent and free R language and its many freely available packages. Funding was provided by Fundación de Investigación Médica Mutua Madrileña and Project TIC2003-09331-C02-02 of the Spanish Ministry of Education and Science (MEC). R.D.-U. is partially supported by the Ramón y Cajal programme of the Spanish MEC. Applications are running on clusters of machines purchased with funds from the RTICCC from the Spanish FIS. Funding to pay the Open Access publication charges for this article was provided by the Fundación de Investigación Médica Mutua Madrileña.

Conflict of interest statement. None declared.

REFERENCES

1. Ge,Y., Dudoit,S. and Speed,T. (2003) Resampling-based multiple testing for microarray data analysis (with discussion). *TEST*, **12**, 1–77.
2. Reiner,A., Yekutieli,D. and Benjamini,Y. (2003) Identifying differentially expressed genes using false discovery rate controlling procedures. *Bioinformatics*, **19**, 368–375.
3. Potter,J.D. (2003) Epidemiology, cancer genetics and microarrays: making correct inferences, using appropriate designs. *Trends Genet.*, **19**, 690–695.
4. Díaz-Uriarte,R. (2005) Supervised methods with genomic data: a review and cautionary view. In Azuaje,F. and Dopazo,J. (eds),

Data Analysis and Visualization in Genomics and Proteomics, chapter 12, Wiley New York, pp. 193–214 .

5. Smyth,G.K. (2004) Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat. Appl. Genet. and Mol. Biol.*, **3**, Article 3.
6. Ambroise,C. and McLachlan,G.J. (2002) Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc. Natl Acad. Sci. USA*, **99**, 6562–6566.
7. Simon, R.,Radmacher,M.D., Dobbin,K. and McShane, L.M. (2003) Pitfalls in the use of dna microarray data for diagnostic and prognostic classification. *J. Nat. Cancer Inst.*, **95**(1), 14–18.
8. Varma,S. and Simon,R. (2006) Bias in error estimation when using cross-validation for model selection. *BMC Bioinformatics*, **7**(1).
9. Dudoit,S. and Fridlyand,J. (2003) Classification in microarray experiments. In Speed, T., (ed.), Statistical Analysis of Gene Expression Microarray Data, chapter 3, pp. 93–158 Chapman & Hall New York.
10. Somorjai, R.L., Dolenko,B. and Baumgartner,R. (2003) Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. *Bioinformatics*, **19**, 1484–1491.
11. Pan, K.H., Lih,C.J. and Cohen, S.N. (2005) Effects of threshold choice on biological conclusions reached during analysis of gene expression by DNA microarrays. *Proc. Natl Acad. Sci. USA*, **102**, 8961–8965.
12. Díaz-Uriarte,R. and Alvarez deAndrés,S. (2006) Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, **7**.
13. Sutter, H. (2005) The free lunch is over: A fundamental turn toward concurrency in software. *Dr. Dobb's Journal*, **30**(3), 202–210.
14. Kontoghiorghes,E.J. (ed.) (2006) *Handbook of Parallel Computing and Statistics*, Chapman & Hall, CRC Press, Boca Raton, FL.
15. Pacheco,P. (1997) *Parallel Programming with MPI*, Morgan Kufman, San Francisco.
16. Hyatt,G., Melamed,R., Park,R., Seguritan,R., Laplace,C., Poirot,L., Zucchelli,S., Obst,R., Matos,M., Venanzi,E. *et al.* (2006) Gene expression microarrays: glimpses of the immunological genome. *Nat. Immunol.*, **7**, 686–691.
17. Rhodes,D.R. and Chinnaiyan, A.M. (2005) Integrative analysis of the cancer transcriptome. *Nat. Genet.*, **37 Suppl**, S31–S37.
18. Dudoit,S., Gentleman, R.C. and Quackenbush,J. (2003) Open source software for the analysis of microarray data. *Biotechniques*, **Suppl**, 45–51.
19. Baxter,S.M., Day,S.W., Fetrow,J.S. and Reisinger,S.J. (2006) Scientific software development is not an oxymoron. *PLoS Comput. Biol.*, **2**, e87+.
20. Fogel,K.F. (2005) *Producing Open Source Software*, O'Reilly, Sebastopol, CA.
21. Olshen,A.B., Venkatraman,E.S., Lucito,R. and Wigler,M. (2004) Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, **5**, 557–572.
22. Hsu,L., Self,S.G., Grove,D., Randolph,T., Wang,K., Delrow,J.J., Loo,L. and Porter,P. (2005) Denoising array-based comparative genomic hybridization data using wavelets. *Biostatistics*, **6**, 211–226.
23. Price,T.S., Regan,R., Mott,R., Hedman,A., Honey,B., Daniels,R.J., Smith,L., Greenfield,A., Tiganescu,A., Buckle,V. *et al.* (2005) SW-ARRAY: a dynamic programming solution for the identification of copy-number changes in genomic DNA using array comparative genome hybridization data. *Nucleic Acids Res.*, **33**, 3455–3464.
24. Lingjaerde, O.C., Baumbusch, L.O., Liestol, K., Glad, I.K. and Borresen-Dale, A.L. (2005) CGH-Explorer: a program for analysis of array-CGH data. *Bioinformatics*, **21**, 821–822.
25. Dave,S.S., Wright,G., Tan,B., Rosenwald,A., Gascoyne,R.D., Chan,W.C., Fisher, R.I., Braziel,R.M., Rimsza,L.M., Grogan,T.M. *et al.* (2004) Prediction of Survival in Follicular Lymphoma Based on Molecular Features of Tumor-Infiltrating Immune Cells. *N. Engl. J. Med.*, **351**, 2159–2169.
26. Psarros,M., Heber,S., Sick,M., Thoppae,G., Harshman,K. and Sick,B. (2005) RACE: Remote Analysis Computation for gene Expression data. *Nucleic Acids Res.*, **33**, W638–W643.
27. Romualdi,C., Vitulo,N., Favero,M.D. and Lanfranchi,G. (2005) MIDAW: a web tool for statistical analysis of microarray data. *Nucleic Acids Res.*, **33**, W644–W649.
28. Montaner,D., TÃrraga,J., Huerta-Cepas,J., Burguet,J., Vaquerizas,J.M., Conde,L., Minguez,P., Vera,J., Mukherjee,S., Valls,J. *et al.* (2006) Next station in microarray data analysis: Gepas. *Nucleic Acids Res.*, **34**(Web Server issue), W486–W491.
29. Rainer,J., Sanchez-Cabo,F., Stocker,G., Sturn,A. and Trajanoski,Z. (2006) Carmaweb: comprehensive r- and bioconductor-based web service for microarray data analysis. *Nucleic Acids Res.*, **34**(Web Server issue), W498–W503.
30. Willenbrock,H. and Fridlyand,J. (2005) A comparison study: applying segmentation to array cgh data for downstream analyses. *Bioinformatics*, **21**, 4084–4091.
31. Lai,W.R.R., Johnson,M.D.D., Kucherlapati,R. and Park, P.J.J. (2005) Comparative analysis of algorithms for identifying amplifications and deletions in array cgh data. *Bioinformatics*, **21**, 3763–3770.
32. Herrero,J., Al-Shahrour,F., Díaz-Uriarte,R., Mateos, Á., Vaquerizas,J.M., Santoyo,J. and Dopazo,J. (2003) GEPAS, a web-based resource for microarray gene expression data analysis. *Nucleic Acids Res.*, **31**, 3461–3467.
33. Herrero,J., Vaquerizas,J.M., Al-Shahrour,F., Conde,L., Mateos,Á., Santoyo,J., Díaz-Uriarte,R. and H.Dopazo,J. (2004) New challenges in gene expression data analysis and the extended GEPAS. *Nucleic Acids Res.*, **32**, W485–W491.
34. Vaquerizas,J.M., Conde,L., Yankilevich,P., Cabezon,A., Minguez,P., Diaz-Uriarte,R., Al-Shahrour,F., Herrero,J. and Dopazo, J. (2005) GEPAS, an experiment-oriented pipeline for the analysis of microarray gene expression data. *Nucleic Acids Res.*, **33**, W616–W620.
35. Díaz-Uriarte,R., Al-Shahrour,F. and Dopazo,J. (2003) The Use of Go Terms to Understand the Biological Significance of Microarray Differential Gene Expression Data, pp. 233–247 in Methods of Microarray Data Analysis III, papers from Camda '02, Kluwer.
36. Al-Shahrour,F., Díaz-Uriarte,R. and Dopazo,J. (2004) FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics*, **20**, 578–580.
37. Al-Shahrour,F., Díaz-Uriarte,R. and Dopazo,J. (2005) Discovering molecular functions significantly related to phenotypes by combining gene expression data and biological information. *Bioinformatics*, **21**(13), 2988–2993.
38. Alibés,A., Yankilevich,P., Cañada,A. and Diaz-Uriarte,R. (2007) Idconverter and idclight: conversion and annotation of gene and protein ids. *BMC Bioinformatics*, **8**, 9.
39. R Development Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing Vienna, Austria (2004).
40. Vaquerizas,J.M., Dopazo,J. and Díaz-Uriarte,R. (2004) DNMAD: web-based diagnosis and normalization for microarray data. *Bioinformatics*, **20**, 3656–3658.
41. El-Ansary,S., Grolaux,D., Van Roy,P. and Rafea,M. (2005) Overcoming the multiplicity of languages and technologies for web-based development using a multi-paradigm approach. In Van Roy, P., (ed.), *Multiparadigm Programming in Mozart/OZ*, chapter 10, Springer, pp. 113–124.